

Optical encoder controls range switch

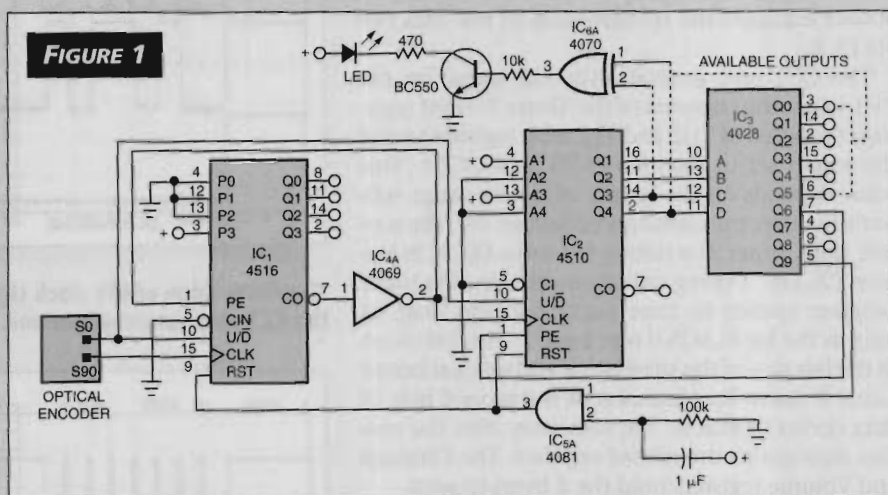
W DIJKSTRA, WAALRE, THE NETHERLANDS

Instead of using a counter-controlled, pushbutton-activated range switch, you can use an optical encoder. Inexpensive encoders are available, and they occupy minimal space on the front panel of an instrument. Moreover, an encoder gives you the opportunity to select the optimum operating speed. However, at positions near the transition points in counter position, mechanical shocks can provoke false switching. The circuit in Figure 1 overcomes the false-switching problem.

Output S0 of the optical encoder controls the up/down inputs of counters IC₁ and IC₂. Output S90 connects to the clock input of the HEF4516 binary counter (IC₁). When this counter reaches 0 or 15, the output CO goes low and clocks (via an inverter) the HEF4510 decimal counter (IC₂).

Simultaneously, the binary counter assumes a value of eight. Thus, it requires eight pulses of the optical encoder to alter the position of the decimal counter. You must stop turning the optical encoder within eight pulses, which in practice is eminently possible. When you want more security, you can feed the outputs Q3 and Q4 of the decimal counter to an exclusive OR gate. When the output of the XOR gate is high, changing the state of the decimal counter requires a minimum of four pulses from the optical encoder.

The circuit provides control with bidirectional hysteresis.



An optical encoder, immune to false switching, takes the place of a counter-controlled range switch.

To stop the decimal counter at zero when counting down, the counter resets when output Q9 of the 1-of-10 decoder HEF4028 (IC₃) goes high. To limit the number of decoder positions, you can load the decimal counter one position lower than the maximum output position you want to reach with the output decoder. The configuration in Figure 1 loads the decimal counter with the value five when Q6 (the seventh position) of IC₃ goes high. (DI #2255).

To Vote For This Design, Circle No. 399

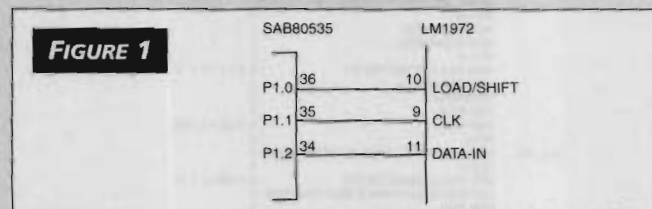
μC controls digital potentiometer

LUKASZ SLIWczynski, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

Many digitally controlled potentiometers (for example, the LM1971/2/3 from National Semiconductor, www.national.com) incorporate a three-wire serial digital interface, using data, clock, and enable lines. In Figure 1, the potentiometer's nomenclature for these lines is Data-In, Clk, and Load/Shift, respectively. The assembler program in Listing 1 provides an interface to an SAB80535 μC. The main idea of the method is to use the capture/compare capability of Timer 2 in the μC to provide the timing relationship between the Data-In and Clk signals. The principal control of the interface comes from subroutine S16BIT in Listing 1.

To program the potentiometer, the μC must send 2 bytes to it—the "channel address," followed by the attenuation value with its most significant bit first. Sending a byte starts

by loading the number of bits to send into the μC's BCOUNT register and initiating the P1 lines (setting P1.0 through P1.3 to a low state). Next, Timer 2 starts with overload enabled. The routine sets two digital compare/capture units (CC1 and



The rising edge of the clock signal validates data loading into the Data-In and Load/Shift pins of the potentiometer.

CC3 in the μ C) to the "compare" mode by writing 88H into the CCEN register. The contents of the register decrease after each interrupt. In this way, eight consecutive interrupts occur, each to send 1 bit of data. The interrupt subroutine at address 006BH manages the transmission of the data bits via P1.2.

The CC1 unit generates the Clk signal on pin P1.1 when the contents of the Timer 2 count register (composed of TH2 and TL2 8-bit registers) equal the values set in registers CCL1 and CCH1. This value depends on the length of the interrupt subroutine. After transmission of the last bit, the routine stops Timer 2 by setting the value 0ECH in register T2CON. During data transmission, the main program spends its time waiting in the loop, as long as the bit FLAG5.0 is at logic 1. This bit clears in the last pass of the subroutine and sets just before Timer 2 starts. Transmission of the second byte of data occurs in exactly the same way after the routine reprograms the related registers. The Channel and Volume registers hold the 2 bytes to send.

Figures 2 and 3 show the timing relationships of the interface. In Figure 2, the 2 bytes Channel and Volume are 05AH and 081H, respectively. Data is valid on the rising edge of the Clk signal. Figure 3 shows the time dependence in the interrupt-routine calls and the Clk rising edges. In this design, it takes approximately 740 μ sec to program 2 bytes into the potentiometer with an 8-MHz clock frequency (a 1.5- μ sec machine-cycle time). (DI #2256).

EDN

To Vote For This Design, Circle No. 400

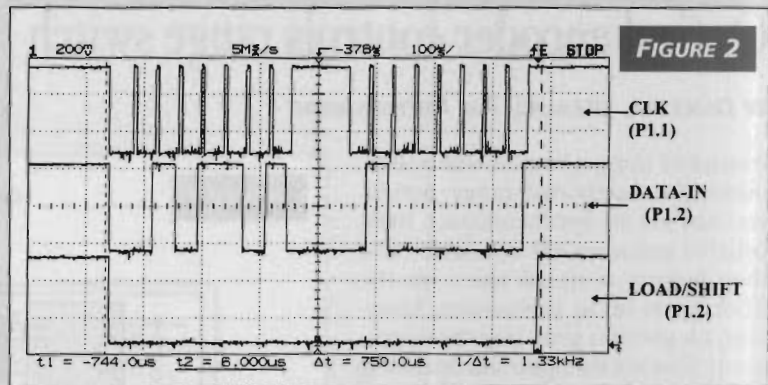


FIGURE 2

The rising edge of the clock signal appears approximately 32 μ sec after the CC3 compare/capture unit in the μ C generates an interrupt.

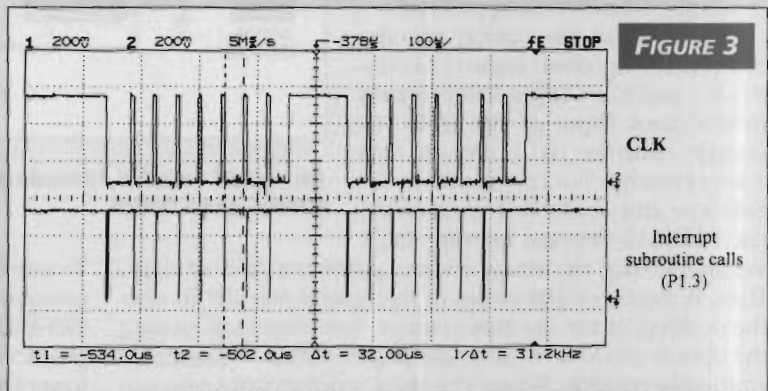


FIGURE 3

The subroutine in Listing 1 provides a simple serial digital interface to National's (and others') digital potentiometers.

LISTING 1—S16BIT SUBROUTINE

```

*****REGISTERS ADDRESSES
DEFINITIONS*****
EQU FLAG5,002FH
EQU BCOUNT,004FH
EQU CHANNEL,004DH
EQU VOLUME,004EH

*****INTERRUPT SUBROUTINE
HANDLER*****
;THE NUMBER OF MACHINE CYCLES TO EXECUTE THE COMMAND ARE
GIVEN AT THE RIGHT-HAND SIDE. THREE ADDITIONAL CYCLES ARE ADDED
FOR THE RESPONSE TO THE INTERRUPT REQUEST

.ORG 006BH
CLR IEN0.7; DISABLE ALL INTERRUPTS      1
PUSH ACC;                               2
PUSH PSW;                               2
MOV A,BCOUNT;                           1
JZ ALL_BY; TEST IF THE LAST BIT TO SEND 2 -> 8 + 3 = 11
DEC A;                                   1
MOV BCOUNT,A;                           1
MOV A,CHANNEL;                           1
RLC A;                                   1
MOV P1.2,C; SEND THE BIT                 2 -> 14 + 3 = 17
MOV CHANNEL,A;                           1
SETB IEN0.7;                             1
POP PSW;                                 2
POP ACC;                                 2 -> 22 + 3 = 25
RETI;                                    2
ALL_BY: MOV A,CHANNEL; THE LAST BIT TO SEND 1
RLC A;                                   1
MOV P1.2,C; SEND THE BIT                 2 -> 12 + 3 = 15
CLR FLAG5.0; INFORM S16BIT FUNCTION      1
POP PSW;                                 2
POP ACC;                                 2
ANL T2CON,#0E0H; STOP TIMER 2            1
SETB IEN0.7; ENABLE INTERRUPTS          1
RETI;                                    2 -> 22 + 3 = 25

```

*****THIS COMMANDS SHOULD BE PLACED SOMEWHERE IN THE BEGINNING OF THE MAIN PROGRAM

```

MOV SP,#05H; SET THE STACK POINTER
MOV IP0,#020H; SET THE INTERRUPT PRIORITY LEVEL OF CC3
MOV FLAG5,#00H; CLEAR ALL BITS

```

*****LM1972 INTERFACE CONTROLLING FUNCTION*****

```

S16BIT: MOV TL2,#(255 - 25);SET TIMER 2 COUNT REGISTER
        MOV TH2,#0FFH
        MOV CCL,#(255 - 25);SET TIMER 2 RELOAD REGISTER
        MOV CCH,#0FFH
        MOV CCL1,#0FCH; SET VALUE FOR CC1 - CLK SIGNAL
        MOV CCH1,#0FFH;
        MOV CCL3,#(255 - 25); SET THE BEGINNING OF INTERRUPT
        MOV CCH3,#0FFH
        MOV BCOUNT,#07H; NUMBER OF BITS TO SEND - 1
        SETB FLAG5.0
        ANL P1.#070H; INITIATE PORT 1
        MOV CCEN,#088H; ENABLE COMPARE MODE FOR CC1 AND CC
        SETB IEN1.5; ENABLE INTERRUPT FROM CC3
        ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
        JB FLAG5.0,S16_1; WAIT UNTIL THE FIRST BYTE IS SEND
        MOV TL2,#(255 - 25); REINITIATE TIMER 2 COUNT REGISTER
        MOV TH2,#0FFH
        MOV BCOUNT,#07H; REINITIATE NUMBER OF BITS TO SEND
        MOV A,VOLUME; SWAP VOLUME AND CHANNEL CONTENTS
        MOV CHANNEL,A
        SETB FLAG5.0
        ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
        JB FLAG5.0,S16_2; WAIT UNTIL THE SECOND BYTE IS SEND
        MOV CCEN,#00H; DISABLE CC1 AND CC3 UNITS
        CLR IEN1.5; DISABLE CC3 INTERRUPT
        ORL P1.#070H; SET OUTPUT PINS TO THE HIGH STATE
S_END: RET

```


Timer inputs double as interrupt-request lines

SK SHENOY, NAVAL PHYSICAL AND OCEANOGRAPHIC LABORATORY, KOCHI, INDIA

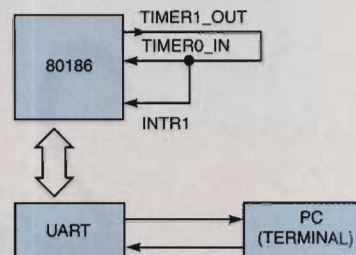
The Intel 80186 is a highly integrated 16-bit μ P that is common to embedded applications. This μ P's built-in interrupt controller has four interrupt-related pins that you can configure in various ways to achieve the maximum of four maskable interrupt-request lines.

In applications that require more than four interrupt-request lines, the only way out is to add an external interrupt controller, such as the 8259A. However, an alternative approach exists: You can through proper programming make the 80186's two timer-input pins function like normal edge-triggered interrupt request lines. The only penalty is an additional latency of about 1 μ sec for an 8-MHz CPU, which is insignificant for many applications. This pseudo interrupt-request line is useful for interfacing any device (UART, DMA controller, coprocessor, or DSP processor) that works with internally vectored, edge-triggered interrupts.

The idea is based on the fact that you can program the timer in a mode during which the timer count resets and then the timer starts counting on a 0-to-1 transition on its input pin. Further, you can set the timer to interrupt when it reaches the value set in its Max Count register. Thus, if the Max Count setting equals 1, the timer generates an interrupt immediately after one timer-clock period from the 0-to-1 transition on the input pin. For an 8-MHz system, with the timer using the internal clock (which is one-fourth the CPU clock), the interrupt time is 500 nsec. The μ P can serve the timer interrupt as any other normal interrupt would. The μ P can also selectively mask and unmask the interrupt using the timer/interrupt-control registers.

You can download a demo program from EDN's Web site,

FIGURE 1



With the proper programming, the Timer0_In input serves as a pseudo interrupt-request line.

www.ednmag.com. (At the registered-user area, go into the Software Center to download the file from DI-SIG, #2277.) The program demonstrates the use of the Timer0_In line as a pseudo interrupt-request line. The program, along with the setup in Figure 1 (which uses the Timer0_out line to generate the interrupt), also compares the latency of this pseudo interrupt with the latency of the normal interrupt line, Intr1. The program was written and compiled using Intel's IC86 compiler and was tested on an 8-MHz 80186 system. The same technique should work for μ Ps and μ Cs that have similar timer capabilities. (DI #2277)

EDN

To Vote For This Design, Circle No. 401

DSP algorithm measures frequency and damping

OLGA BELOUSOVA, LOS ALAMOS, NM, AND ALEXANDER BELOUSOV, NEW YORK, NY

The time-domain DSP algorithm described here allows you to measure the key parameters—natural frequency and damping—in linear, second-order electromechanical systems. The method applies to a range of electromechanical transducers (electromagnetic or electrostatic), including dynamic speakers, seismic geophones, micromachined sensors, and other systems. The algorithm is based on the integral transforms of the terminal voltage of the transducers in a free transient mode (after application and removal of the step-function stimulus). It provides high immunity to both electrical noise and mechanical vibration.

Compared with traditional FFT methods, the algorithm significantly simplifies the computational task, provides better resolution in locating the spectral peak (which corre-

sponds with the natural resonant frequency of the transducer), and allows you to calculate the damping coefficient (which you can not directly extract from an FFT). Figure 1 uses an electromagnetic transducer, stimulated by a step-current function, with sequential integration of its terminal voltage in a free transient mode. The general equation (assuming zero initial phase) is:

$$v(t) = V_0 \cdot \exp(-\omega_0 \beta t) \cdot \sin(\omega_0(1 - \beta^2)^{0.5} t),$$

where V_0 is the final amplitude of the terminal voltage, β is the damping coefficient (an unknown value), and ω_0 is the natural angular resonant frequency of the transducer system. The algorithm is based on the following integral transforms of the terminal voltage, $v(t)$:

$$J_1 = \int_0^{\infty} v(t) dt,$$

$$J_2 = \int_0^{\infty} |v(t)| dt,$$

$$J_3 = \int_0^{\infty} (v(t))^2 dt.$$

Omitting the intermediate math, you can write the computational formulas for β and ω_0 as

$$\beta = (1 + \pi^2 (\ln((J_2 + J_1)/(J_2 - J_1)))^{-2})^{-0.5},$$

$$\omega_0 = 4\beta \cdot J_3 / J_1^2.$$

For a practical implementation in DSP form, you must substitute the indefinite integrals J_1 through J_3 with corresponding finite sums S_1 through S_3 . The ADC must digitize the terminal voltage during a period of time long enough to allow the free transient to settle. Then, the DSP must calculate the three finite sums using the following equations:

$$S_1 = \sum (v_i), S_2 = \sum |v_i|, \text{ and } S_3 = \sum (v_i)^2.$$

The final computational formulas are as follows:

$$\beta = (1 + \pi^2 (\ln((S_2 + S_1)/(S_2 - S_1)))^{-2})^{-0.5},$$

$$\omega_0 = (4\beta \cdot S_3 / S_1^2) / \Delta t,$$

where Δt is the sampling period.

The circuit in **Figure 1** consists of the transducer under test, shown as inductor L_C in series with the coil resistance, R_C ; a current-stimulus circuit (analog switch S with a current-

limiting resistor, R_S , and the integrated ADC. We chose the low-power 10831 ADC from National Semiconductor (www.national.com) because of its convenient serial interface to the μC . Zener diodes D_1 and D_2 (approximately 5V breakdown) protect the ADC's input against overvoltage from inductive spikes).

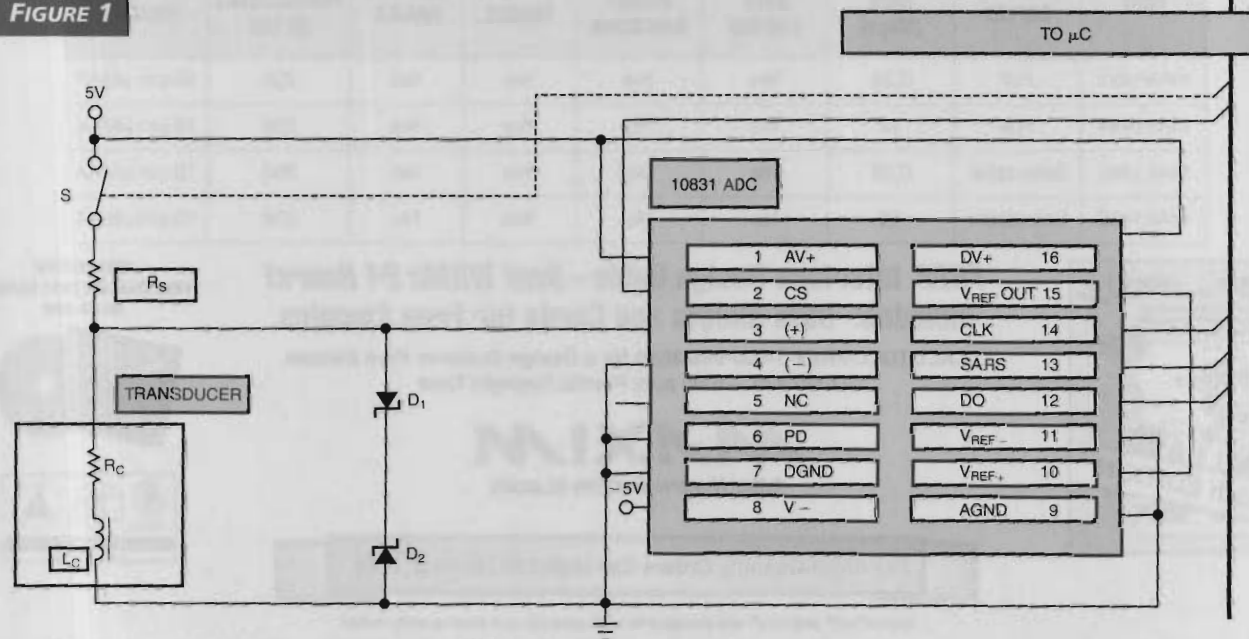
The principal of operation is simple. First, you apply the current step function to the transducer. The duration of the current stimulus should be long enough to allow the transient to settle; this value depends on the estimated natural frequency and damping in the system. For example, for a typical seismic geophone with $f_0=10$ Hz and $\beta=0.6$, the step function should last 0.5 to 1 sec. The same rule applies to the measuring cycle in a free transient. In general, you can stop the measurement when the terminal voltage drops to less than 1 LSB in the chosen ADC. The sampling period, Δt , should be small enough to avoid methodical errors that accrue from substituting the analog integral transforms with discrete sums.

Because the resonant frequencies of mechanical systems are typically low, it is relatively easy to avoid the methodical errors. For example, the maximum sampling rate of the ADC 10831 is 74 kHz; thus, the resulting errors are low. You could also use a 10-bit 10732 ADC, a differential-input, single-supply device. For higher frequency and better resolution, you could use a 12-bit 12130 ADC, which has 14- μ sec throughput time. You can use any embedded-system μC for the method. (DI #2244)

EDN

To Vote For This Design, Circle No. 402

FIGURE 1



A time-domain DSP algorithm measures natural frequency and damping in electromechanical systems.

Video equalizer sharpens VCR images

WAYNE SWARD, CONSULTANT, BOUNTIFUL, UT

Video signals from a VCR lose sharpness, resulting in a flat-looking image. The effect is especially noticeable when copying from one tape to another, as you do in a video-editing system. High-quality recorders minimize this effect, but they are expensive.

The circuit in **Figure 1** sharpens picture images without introducing the shadows, ringing, and noise often observed with commercial video equalizers. The key to this improvement is the coaxial delay line, which replaces the typical LC network in other equalizers.

The circuit applies the video signal to a MAX466 quad video amplifier. The first amplifier drives a sample of the original video into a resistive summing network and a second amplifier through a 3.58-MHz trap. The trap deletes the chrominance-color signal from the original video. The second amplifier applies the remaining luminance-brightness video signal to the shorted coaxial delay line. This amplifier also applies a sample of the luminance signal to the resistive summing network. The total round-trip delay of the delay line is 65 nsec, about half the typical rise time of a VCR output for typical VHS or 8-mm VCRs.

The output of this delay line drives a third video amplifier, which recovers the edges of picture images. The enhanced-edge output of this third amplifier combines some of the original video signal from the first amplifier, some of the

luminance video signal from the second amplifier, and the enhanced-edge video signal from the third amplifier into an improved video signal.

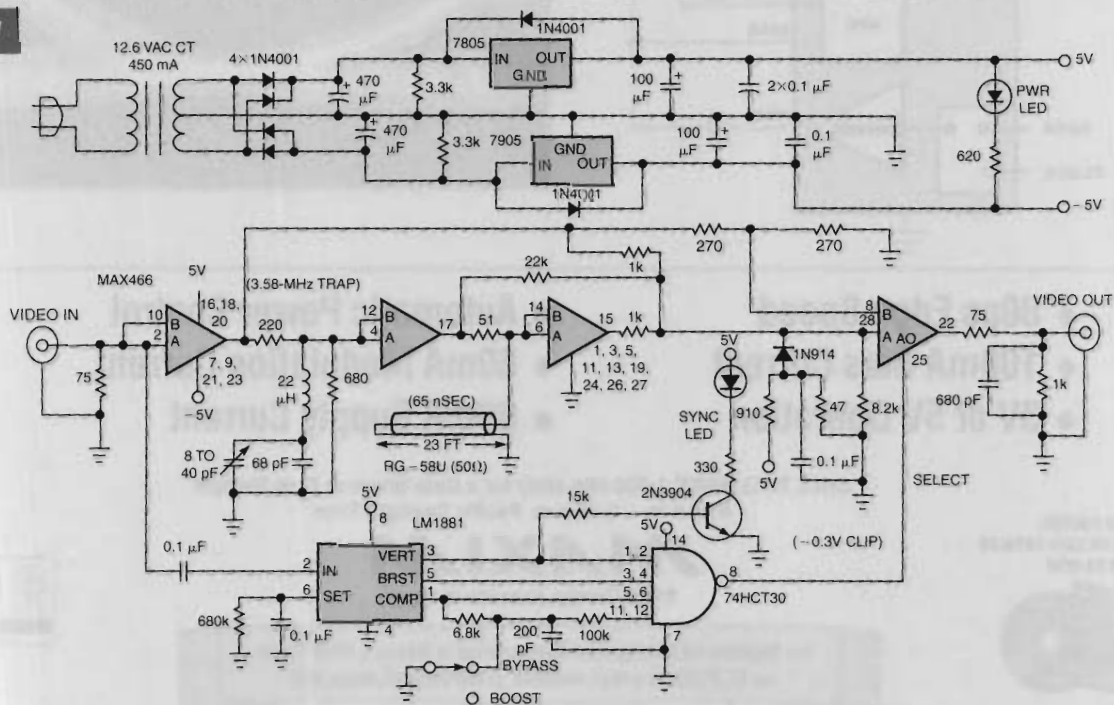
The 1N914 diode and associated resistors clip the video to -0.3V , which differentiates the enhanced video from being confused with normal sync signals. The resulting video signal drives one input of the fourth video amplifier. The second input of the fourth video amplifier is the original video signal, including sync and color burst. The LM1881 video-sync detector and the 74HCT30 NAND gate-switch this fourth video amplifier so that the original sync and color burst from the first amplifier go directly to the output, and the enhanced video signal from the summing network goes to the output during active video time.

The 680-pF capacitor across the video output suppresses switching transients generated in the fourth amplifier when switching between sync/color burst and enhanced video but does not degrade the enhanced video. The circuit also contains a power LED, a vertical-sync-indicator LED, and a bypass/boost switch. The power supply is a conventional linear design. The circuit is built on a double-sided copper-clad pc board. (DI #2251)

EDN

To Vote For This Design, Circle No. 403

FIGURE 1



Using a delay line in place of the typical LC equalizer network allows this circuit to sharpen VCR images without introducing the artifacts commonly produced by commercial equalizers.

Programmable charger fills capacitor banks

DENNIS FEUCHT, INNOVATIA LABS, TOWNVILLE, PA

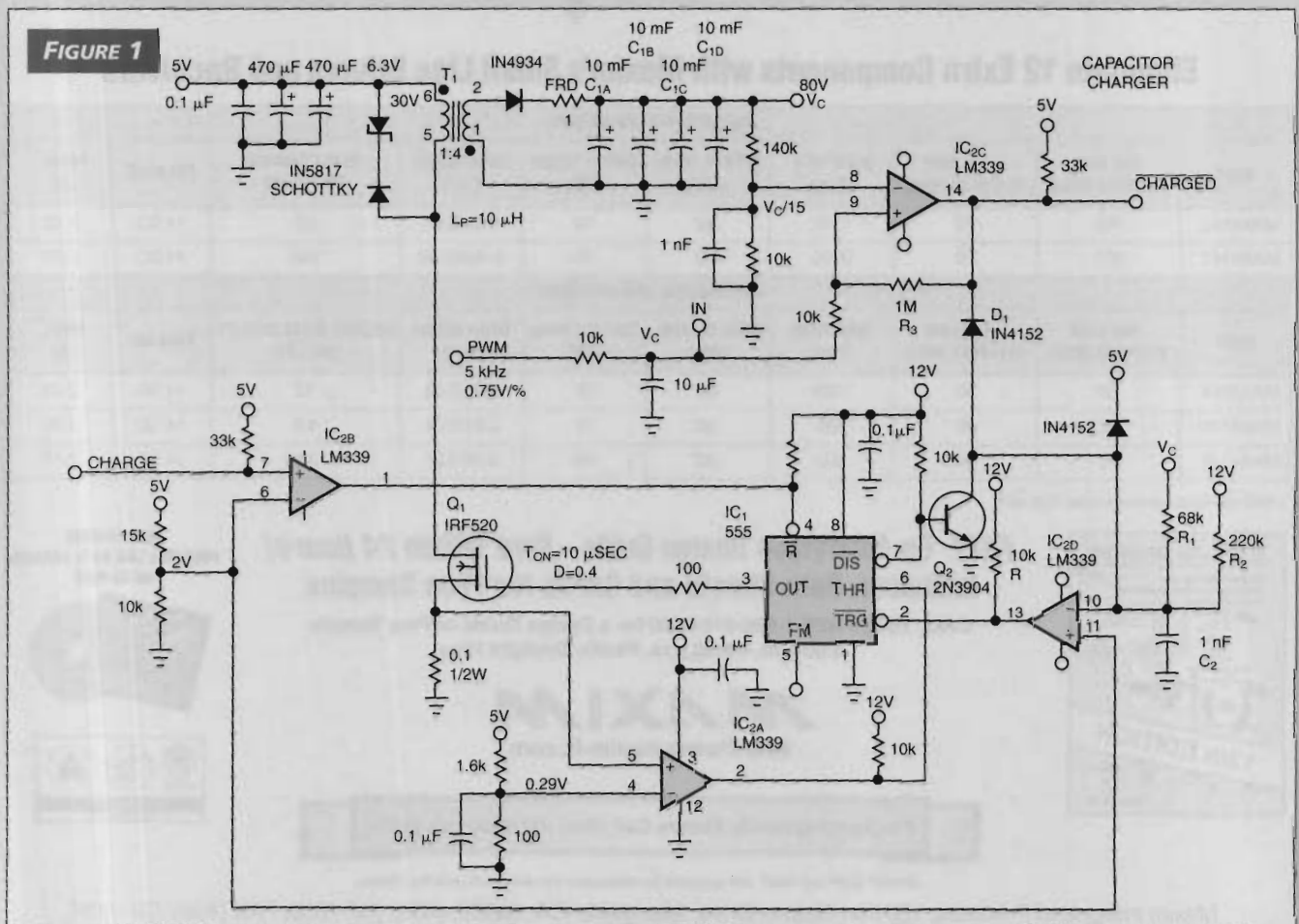
Designed for the use with a pyrotechnic pulse generator, the charger in Figure 1 uses low-cost components to charge the C_1 capacitor bank to a voltage as high as 75V, or to 25V in a few seconds. A 555 timer, IC_1 , controls a flyback converter via the Q_1 transistor switch. The converter draws its charging current from a 5V supply. The 555 turns Q_1 on until T_1 's primary current reaches approximately 3A, as detected by comparator IC_{2A} , which turns IC_1 off. Then, Q_2 also turns off, and C_2 charges through R_1 and R_2 , thereby setting the off-time of the charger cycle. IC_{2D} starts the new cycle by triggering IC_1 .

Because R_1 connects to the charger output (V_C), as C_1 charges, the off-time decreases. The secondary conduction time decreases as V_C opposes T_1 's flux. But the flyback converter transfers no charge when T_1 is not conducting current. To minimize this no-charge time, the off-time decreases with V_C . R_2 is necessary for start-up, when V_C is 0V and cannot

charge C_2 . A digital input (charge) turns the charger on and off.

The target voltage on C_1 depends on the duty cycle of a 5V CMOS-level pulse applied to the PWM input. A 5-kHz PWM waveform has a scale factor of 0.75V/%. A 100% duty cycle produces a 75V target voltage. IC_{2C} detects when this voltage is reached and turns the charger off through D_1 . IC_{2C} 's output (Charged) can also indicate the completion of charging. If you use this output as a computer input, you should monitor it in a loop, because it is unasserted during recharge. R_3 provides hysteresis, and, as C_1 leaks charge and V_C decreases, IC_{2C} changes output state and charging commences, to keep C_1 "topped off." (DI #2260).

To Vote For This Design, Circle No. 404



The duty cycle of a CMOS pulse train determines the target charged voltage for a bank of high-energy capacitors.

μ C provides wireless keypad control

LLOYD KHUC, MOTOROLA INC, AUSTIN, TX

The circuit in **Figure 1** is a simple, 4×4 keypad remote-control system. A 68HC705J1A μ C, IC₁ (**Figure 1a**), costs less than \$1 and controls the keypad functions. When you depress any key, the μ C provides a 4-bit hex-data output and then enables a latch signal to latch the data into IC₂. Next, the μ C enables IC₂ to transmit the signal to the IC₃ encoder to convert the 4-bit hex data to serial data to send to IC₄. IC₄, the RF data-transmitter module, mixes the 315-MHz carrier frequency to transmit.

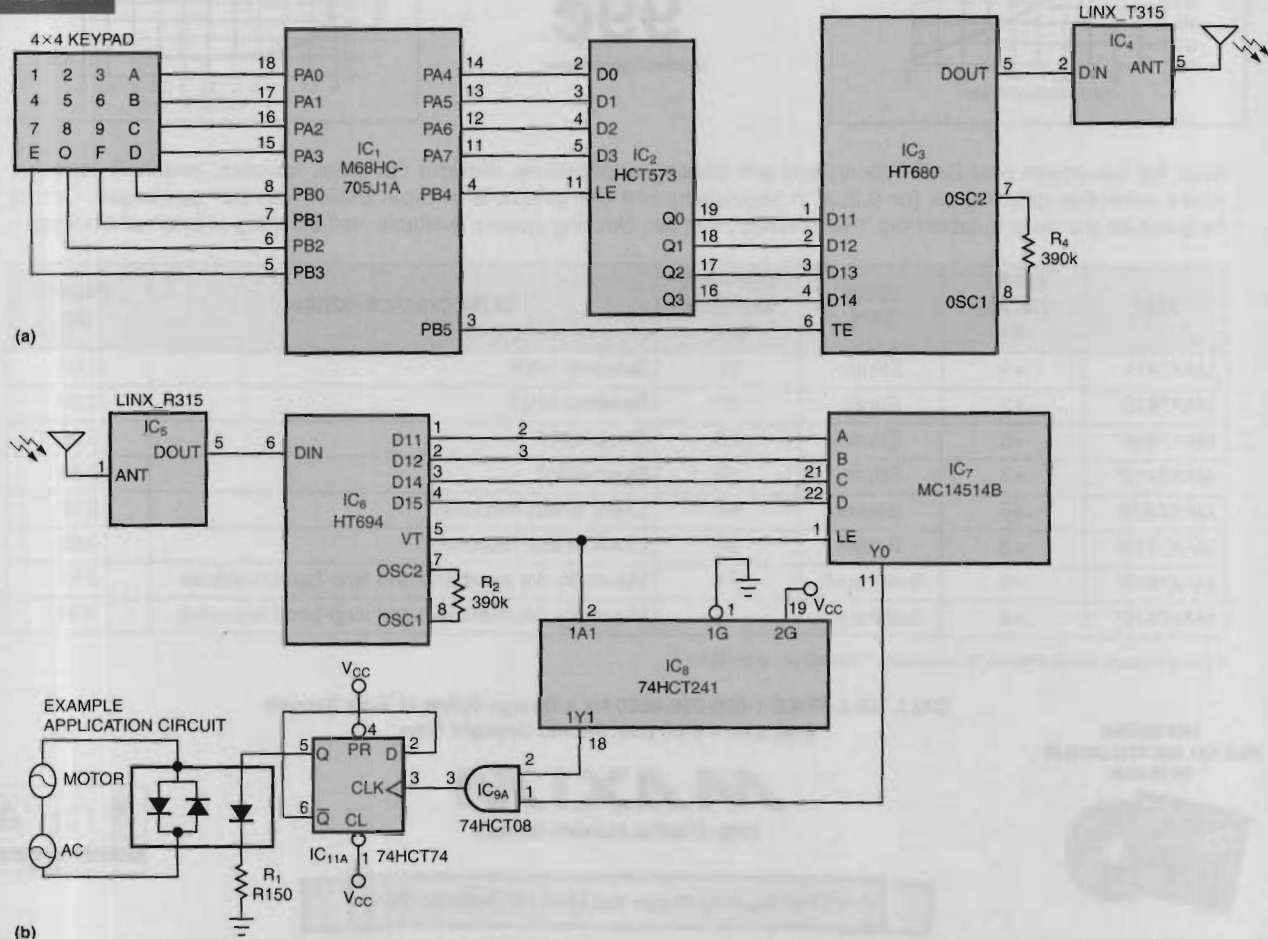
In the receiver circuit (**Figure 1b**), the IC₅ data-receiver module removes the 315-MHz carrier frequency from the receiver signal, and IC₆ decodes the serial data into a 4-bit hex

parallel data output. IC₇ converts the 4-bit hex data into 16 data bits to control 16 application circuits. IC₃ and IC₆ are a matching encoder/decoder pair that eliminates any unwanted interference frequencies. IC₄ and IC₅ are a matching RF data-transmitter/receiver pair with a 315-MHz carrier frequency. You can download the assembler code for the μ C system from EDN's Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2261. (DI #2261).

EDN

To Vote For This Design, Circle No. 405

FIGURE 1



A simple 4×4 keypad gives you wireless control of 16 lines, using an inexpensive μ C, a few logic blocks, and a transmitter/receiver module pair.